The Journal of Supercomputing (2023) 79:3392–3411
https://doi.org/10.1007/s11227-022-04783-y

An improved anomaly detection model for IoT security using decision tree and gradient boosting

Maryam Douiba[1] · Said Benkirane[1] · Azidine Guezzaz[1] · Mourade Azrour[2]

Accepted: 22 August 2022 / Published online: 3 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Internet of Things (IoT) represents a massive deployment of connected, intelligent devices that communicate directly in private, public, and professional environments without human intervention. The increasing number and mobility make them more attractive to attackers. Therefore, many techniques have been integrated to secure IoT, such as authentication, availability, encryption, and data integrity. Intrusion detection systems (IDSs) are an effective security tool that can be enhanced using machine learning (ML) and deep learning (DP) algorithms. This paper presents an improved IDS using gradient boosting (GB) and decision tree (DT) through the open-source Catboost for IoT Security. The proposed model has been evaluated under the improved NSL- KDD, IoT-23, BoT-IoT, and Edge-IIoT datasets using the GPU to enhance the experimental setting. Compared with the well-existed IDS, the results prove that our approach gives good score performance metrics of ACC, recall, and precision, around 99.9% on a record detection and computation time.

Keywords ML · Gradient Boosting · Decision Tree · GPU · IoT Security · Intrusion Detection

1 Introduction.

IoT environments are rapidly spread due to the growth of connected objects and heterogeneous physical devices equipped with various sensors, actuators, and processors. They can exchange information directly or via the Internet without human intervention [1, 5]. An aggregator is an important IoT element and is considered a middleware that connects and manages all heterogeneous devices in IoT

✉ Azidine Guezzaz
a.guzzaz@gmail.com

[1] Technology Higher School Essaouira, Cadi Ayyad University, Essaouira, Morocco

[2] IDMS team, Faculty of Sciences and Technics, Moulay Ismail University of Meknès, Meknes, Morocco

environments [2]. The cloud is an essential component of IoT that represents the most common compute and storage resources of data gathered within a huge amount of devices [3, 5]. Hence, the expansion of IoT can be seen because of its availability and the increasing deployment in various areas such as healthcare systems, smart cities, smart homes, intelligent transportation, and industries [2, 17].

Several works proposed different IoT architectures. The most frequently used is three layers architecture [1, 3, 5], which is still not adequate for the current development of IoT. Five-layer architecture [3, 5] consists of the perception layer, composed of devices, sensors, and actuators, and is used to collect data from sensors and actuators of the IoT environment. The transport layer manages communication between devices and transfers collected data to the processing layer, which is responsible for storing, analyzing, and processing huge amounts of data. Also, it provides service to the lower layers. The application layer delivers brilliant service to users. The business layer manages the whole IoT system. The cloud- and fog-based IoT is a contemporary architecture that combines edge computing, fog computing, and cloud computing [3, 5]. As depicted in Fig. 1, the cloud- and fog-based IoT consists of a monitoring level, which monitors power, resources, responses, and services, a processing level that filters and analyzes sensor data, and a storage level, which delivers storage functionalities such as data replication, data distribution, and data storage.

IoT security is characterized by verification, authorization, privacy, access control, information storage, system configuration, and management [2]. The existing methodologies and standards used for IoT security have many issues due to the complexity of the systems and the heterogeneity of devices used. IDS may, nevertheless, be a crucial and highly beneficial security solution for ensuring the IoT network's security [1–3, 13, 14, 48, 55]; it can be deployed in IoT with other security measures, such as encryption techniques, access control, securing routing, and trust manager authentication [8, 12]. In addition, IDS can be categorized into two types: host-based IDS (HIDS) and network-based IDS (NIDS) [6, 7, 10, 15, 32, 53, 54]. Our study focuses on NIDS, network traffic attacks, and sending alerts to the network administrator. It is placed outside the network infrastructure and performs the

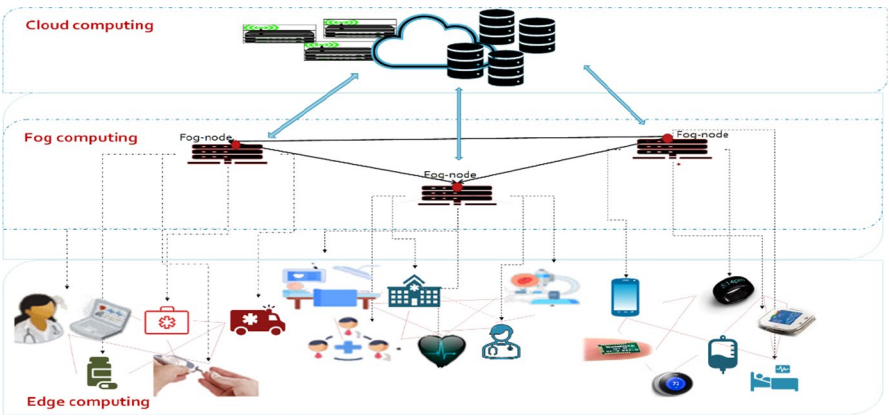

Fig. 1 Cloud- and fog-based IoT architecture

analysis on a copy of the inline network traffic. As a result, actual inline network performance is not affected. It initially checks the packets; it receives from host- or network-based sensors and then utilizes feature extraction to attempt to extract features. The last step is to perform classification algorithms to identify the intrusion or anomaly using retrieved features. Moreover, it is essential to boost IDS with emerging improved artificial intelligence, such as ML and DL [4, 6, 7, 10, 56]. Hence, Intrusion detection is still an ongoing research area because it is a robust approach that allows for secure and protected IoT environments against many attacks such as service scanning, keylogging denial of service (DoS), and distributed DoS (DDoS) [6, 10, 48–50]. A set of ensemble learning, ML, and DL methods have been incorporated to propose enhanced IDS. Even with those efforts, many problems remain to be solved, such as real-time detection, class imbalance, quality improvement, high dimensionality, huge volume, and time performance [5].

The main goal of this work is to solve some intrusion detection limits by improving and enhancing the classification performance. Therefore, we validate an anomaly IDS model using Catboost [42, 43], an efficient open-source package combining GB and DT algorithms. Our contribution is summarized in two essential parts. Our contribution is summarized in two points. The first is to increase the accuracy and precision of IDS, and the second is to reduce the detection time. For that, we used the Catboost algorithm, especially gradient boosting for decision trees and benefitting from a library with multi-GPU implementation support to deal with the huge volume to reduce processing time and detection time. Furthermore, CatBoost allows to deal with the categorical features using CatboostEncoder and solve class imbalance by optimizing the detection of minority classes using target statistics and gradient boosting. We tested the model and provided a comparative study on four datasets NSL-KDD [45], BoT-IoT [46], IoT-23 [40], and Edge-IIoTset [23], to confirm the stability and to determine the effectiveness of our solution. The experimental results prove that our model performs well and makes reliable decisions.

The remainder of this paper is structured as follows. Section 2 reviews some related works in intrusion detection approaches that include ML, DL, and ensemble learning techniques. Section 3 describes the essential steps of the proposed design and suggested solutions to validate our intrusion detection approach. The experimental evaluation and results are discussed in Sect. 4. Finally, the paper is achieved with a conclusion and future works.

2 Related works

This section reviews and cites some recent related works of IDS that integrate ML and DL algorithms for enhancing IoT security.

IoT security is a crucial issue because of the heterogeneity of IoT systems and the insufficiency of security measures embedded in devices [2, 5, 48]. IoT security issues are based on traditional and existing security mechanisms such as authentication, securing routing, encryption, key management protocols, authorization frameworks, IDS, and other approaches [1, 2]. However, they are insufficient to better secure IoT [1, 10]. In addition, the lack of measures considers the limited resource of energy

and memory [3]. IoT architectures are distributed. Hence, sensors and devices need to communicate and aggregate data before getting to the Internet and then connect to the Internet via a smart gateway using user datagram protocol (UDP), transmission control protocol, address resolution protocol, IPv6 Internet Control Message Protocol (ICMP), Internet group management protocol, or Reverse Address Resolution Protocol [3, 16, 17]. On the other hand, intrusion detection is a defense mechanism used to monitor traffic and detect vulnerabilities within the network infrastructure. It can identify and stop malicious activities [6, 7, 51, 52].

In the literature review, as depicted in Table 1, many researchers have investigated their efforts to enhance intrusion detection to protect the IoT environment. Accordingly, Misra et al. [24], and Kasinathan et al. [25] presented novel security architecture for detecting DDoS attacks in IoT. In 2013, Raza et al. [28] created IDS called SVELTE to secure IoT with an integrated mini firewall that uses RPL as a routing protocol in IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) networks. However, in 2015, C. Cervantes et al. [29] benchmarked SVELTE and presented the Intrusion detection of SiNkhole attacks on 6LoWPAN for InterneT of ThIngs (INTI) system for detecting sinkhole attacks on 6LoWPAN for IoT. The simulation result showed that INTI has a low rate of false positives and negatives than SVELTE. In 2016, Sonar et al. [26] proposed an intrusion detection approach to secure IoT against DDoS. They explored the effectiveness of deploying ML and DL algorithms in IDS to improve the security of IoT systems, such as Hodo et al. [11, 13] proposed an ANN IDS model to classify threat analysis of IoT networks. The evaluation of this model achieves over 99% accuracy. In 2017, Fadlullah et al. [9] proposed a background of DL evolving machine intelligence toward intelligent network traffic. A set of ML and IDS contributions for IoT security are analyzed, combining IoT, IDS, and ML. Simultaneously, Diro et al. [20] developed a distributed attack DL detection scheme for IoT security. The model can better detect attacks than centralized ones; the accuracy increased from 96 to over 99%. In 2018, Prabavathy et al. [21] proposed an IDS design of cognitive fog computing for IoT environments. The proposed design is implemented using the OS-ELM algorithm at distributed fog nodes and achieves 97.36% accuracy with a reduced false alarm rate of 0.37%. One year later, Verma et al. [19] compared and brought the performances of many supervised ML algorithms to select a reliable classifier model for IoT security. They proposed an IDSs model based on ensemble learning and proved that Gradient Boosting Machine (GBM) performs best in sensitivity at 99.53%. Furthermore, Chaabouni et al. [18] proposed an OneM2M IDS based on edge ML for IoT security. The experimental results demonstrate good results in detection rate 93.80%, accuracy 92.32%, precision 92.95%, FPR 1.53%, and CPU training time 9280 ms. Al-kasassbeh et al. [31] The LightGBM algorithm achieved almost 100% accuracy, proving this ML algorithm's efficiency over DL strategies. In 2021, Ullah et al. [12] laid out a deep learning model IDS using a convolutional neural network for binary and multi-cast classifications, the model gives the minimum detection rate of around 99.7%. Therefore, from the above-related works, it is clear that robust intrusion detection approaches are achieved using gradient GBM, extreme gradient boosting (XGB), and LightGBM.

Table 1 Classification and comparison study of IDSs for IoT security

	Year	Used Learning method	Detection method	Dataset
Al-kasassbeh et al. [31]	2020	LightGBM	Hybrid-based	–
Misra et al. [24]	2011	–	Learning Automata Detect DDos attack	–
Raza et al. [28]	2013	–	SVELTE distributed IDS Hybrid-based	Simulation with contiki and cooja
Kasinathan et al. [25]	2013	–	Centralized approach for Dos attack	–
Cervantes et al. [29]	2015	–	INIT Centralized approach to detect sinkhole attack	Simulation with Cooja
Sonar et al. [26]	2016	–	Distributed approach to detect Dos attack	Simulation with contiki and cooja
Hodo et al. [13]	2016	Multi-layer perceptron	Anomaly based	–
Hosseinpour et al. [14]	2016	Artificial immune	Distributed anomaly based	KDD-Cup 99 SSH Brute
Diro et al. [20]	2017	Multi-layer deep learning	Distributed anomaly based	NSL-KDD
Bostani et al. [30]	2017	Optimum path forest	Distributed hybrid-based	NSL-KDD
Prabavathy et al. [21]	2018	Online sequential extreme learning	Distributed anomaly based	NSL-KDD
Verma et al. [19]	2019	Random forests, Adaboost, GBM, Extremely randomized trees	Distributed anomaly based	CIDDS-001, UNSW-NB15, NSL-KDD
Chaabouni et al. [18]	2020	Decision tree J48	Distributed anomaly based	OneM2Mdata
Ullah et al. [12]	2021	CNN	Distributed anomaly based	BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, IoT-23

All methods based on gradient boosting are extremely powerful optimization algorithms. Moreover, according to the comparison by Abdullahi et al. [43], Catboost is the most efficient; it outperforms all existing implementations of GBDT, such as GBM, XGB, LightGBM, and H2O. Catboost allows combining all positive points. Hence, implementing GB using binary DT as basic predictors [42, 44] that use the same splitting criterion on a whole level of the tree makes it less prone to overfitting and faster execution at test time [36].

Catboost offers a very efficient way to encode categorical features and has a library with multi-GPU implementation support [42]. In order to evaluate IDSs performance, many datasets are available, for instance, KDD99, UNSW-NB15, Kyoto 2006+, NSL-KDD, BoT-IoT, IoT-23, IoT Network Intrusion, MQTT-IoT-IDS2020, and CICIDS2017 [7, 40, 45, 46]. This evaluation's most commonly used metrics are ACC, recall, FPR, FNR, precision, and f1-score.

3 Optimized intrusion detection model

This section details various solutions to validate our intrusion detection approach for IoT environment security.

3.1 Proposed design

Our contribution aims to propose and implement an optimized model improving detection rate, accuracy, and processing time. The architecture of the proposed model is illustrated in Fig. 2.

This model aims to validate optimized IDS based on the Catboost classifier combining GB and DT algorithms. Therefore, our proposed approach can reduce the gradient estimation bias and improve the generalization capability. The training stage is carried out using GPU. As depicted in Fig. 2, our optimized model process is divided into four essential steps:

- **Step 1** Data pre-processing:
  Data is prepared and understood. Therefore, we identified and removed all inconsistent values, such as real and NaN values.
- **Step 2** Feature engineering:
  The feature vector $(X_m^1, X_m^2, ..., X_m^n)$ and target label $Y_m(y_1, y_2..., y_m)$ are defined and prepared with a Catboost encoder using the average label values on the whole train dataset to reduce the overfitting problem [44]. Then the categorical values X_m^i are encoded with (CatBoost Encoder) by greedily using the TS on the whole dataset to reduce overfitting, avoid target leakage, and normalization problems. Subsequently, the features are transformed and combined. The ordering approach creates a strong predictor in each category based.
- **Step 3** Training and building of the model
  The test and train data are reconstructed as shown in Table 2, and the hyperparameters are identified, such as max depth, iterations, task type, estimation

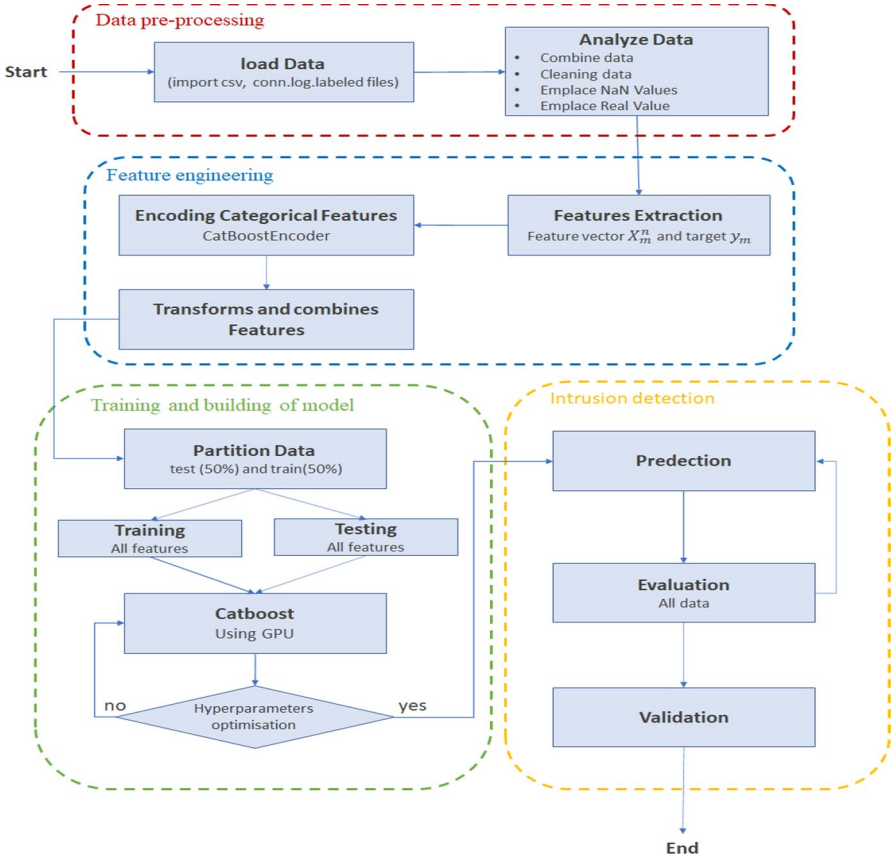

Fig. 2 Proposed design of our IDS approach for IoT security

Table 2 The confusion matrix

Actual		
Normal	True Negative	False Positive
Attack	False Negative	True Positive
	Normal	Attack
	Predicted	

Table 3 Data reconstructions

	Learning		Validation
Datasets	Training (50%)	Testing (50%)	All data (100%)
BoT-IoT	1,834,261	1,834,261	3,668,522
NSL-KDD	12,596	12,596	25,192
IoT-23	673,311	673,311	1,346,622
Edge-IIoT	78,900	78,900	157,800

method, loss function, boosting type, and eval metric. All hyperparameters are optimized to obtain the best performance, as shown in Table 3. The training process is implemented using ML ensemble classifier Catboost and GPU processing.

- **Step 4** Intrusion detection:

The building model can predict an attack as positive. It is evaluated and validated based on metric performances from the confusion matrix (Table 4), such as ACC, recall, precision, FPR, FNR, and f1-score.

3.2 Catboost implementing

Assume that we observe the data D with m samples and n features:

$$D = \{(X_j^i, y_j)\} \; when \begin{cases} i = 1, ..., n \\ j = 1, ..., m \end{cases} \tag{1}$$

The dimensional feature vector $X_J \in \mathbb{R}^m$ and the corresponding label $y_J \in \mathbb{R}$. The symmetric DT is defined in Eq. 2 [44]:

$$h(a) = \sum_{1}^{k} w_k 1_{\{a \in R_k\}} \tag{2}$$

h(a) is constructed by superposition of estimated response features of all regions: R: 1… k. with w_k is the estimated value of the predicted class label of each region k and $1_{\{a \in R_k\}}$ is the indicator function defined in Eq. 3

$$1_{\{a \in \mathbb{R}_k\}} = \begin{cases} 1 \; if \; a \in R_k \\ 0 \; if \; a \notin R_k \end{cases} \tag{3}$$

Table 4 Optimized Catboost hyperparameters

Hyperparameter	Value
max_depth	3
iterations	150
loss_function	Logloss
eval_metric	Accuracy
task_type	GPU
learning_rate	0.9
Custom_loss	AUC, Accuracy, F1 Precision, Recall
Leaf_estimation_method	Gradient

The training on GB aims to minimize expected loss $\mathcal{L}(F) := \boldsymbol{E}L(y, F(x))$ with a smooth loss function $L(.,.)$ and F is the approximate function. So F^t is the series of approximate functions defined in Eq. 4 [44].

$$F^t : \mathbb{R}^m \to \mathbb{R}, F^t = F^{t-1} + \alpha h^t \tag{4}$$

α is a step size and h^t is a base predictor from a family of functions H of Eq. 5 [44].

$$h^t = \arg\min_{h \in H} \mathcal{L}\left(F^{t-1} + h^t\right) = \arg\min_{h \in H} \mathrm{E}\mathcal{L}\left(y, F^{t-1}(x) + h^t(x)\right) \tag{5}$$

This minimization problem is solved by the negative gradient $-g^t(x, y)$ with Eq. 6 [44].

$$g^t(x, y) = \frac{\partial L(y, s)}{\partial s}|_{s=F^{t-1}(x)} \tag{6}$$

h^t is chosen so that $h^t(x)$ approximates $-g^t(x, y)$ the DT function that minimizes expected loss so h^t became from Eq. 7 [44].

$$h^t = \arg\min_{h \in H} \mathrm{E}\left(-g^t(x, y) - h(x)\right) \tag{7}$$

This expectation is approximated by considering dataset D. Moreover, Catboost solves prediction shift by using ordered boosting and categorical features problems with the greedy target statistics (TS). It is an estimate of the expected target y in each category x_j^i with jth training defined in Eq. 8.

$$\hat{x}_j^i = \mathrm{E}\left(y | x^i = x_j^i\right) = \frac{\sum_{k=1}^{n} 1_{\left\{x_k^i = x_j^i\right\}} \cdot y_j + \alpha p}{\sum_{k=1}^{n} 1_{\left\{x_k^i = x_j^i\right\}} + \alpha} \tag{8}$$

When p is set to the average of the target value over the sample with the parameter $\alpha > 0$. According to standard parts of IDS, our proposed approach is designed in four steps: data preprocessing, feature engineering, Training and building of the model, and intrusion detection. It integrates a GB classifier which gives efficient decisions. Indeed, if it is about a binary classification, the most efficient ML is the DT [34, 35]. In practice, datasets include both numerical and categorical features. The problem of categorical features is well solved with the digital conversion that proposes Catboost.

We verified our model using recall, accuracy, and precision. The proportion of correctly recognized samples to the total number of samples is how accuracy is measured. The proportion of correctly categorized items to the total TP (True Positive) and FP is used to gauge precision (False Positive). Calculating the recall value involves dividing the total number of TP measurements by the total number of TP and FN (False Negative). We also calculate FPR and FNR. The FPR (False Positive Rate) is the percentage of normal samples that test positive, while the FNR (False Negative Rate) is the percentage of abnormal samples that test negative.

- True Positive: the model predicts attack as true and is actually true.
- True Negative: the model predicts normal as not true and is actually normal.
- False Positive: the model predicts attack but is actually not.
- False Negative: the model predicts normal but is actually not.

$$Accuracy = \frac{TP + TN}{TP + TP + FP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$FNR = \frac{FN}{FN + TP} \tag{12}$$

$$FPR = \frac{FP}{FP + TN} \tag{13}$$

4 Experimental evaluation and results

4.1 Datasets and simulation setup

The evaluation of IDS is an essential issue. Moreover, the optimal parameters of performance of any classifier depend on the dataset used in the training and the test of the model. In this paper, four datasets are used:

· Edge-IIoTset [23] The dataset was generated using a specially designed IoT/IIoT testbed with a prominent representative set of protocols, sensors, and cloud/edge configurations. Data is generated from several sensors, such as humidity, temperature, water level, heart rate, pH, etc.

· BoT-IoT [46] is evolved and labeled for possible multiclass purposes. The label features indicated an attack flow, the attacks category, and the subcategory. BoT-IoT has a more significant number of attacks, 99.99%, than benign ones, 0.01%, and it has a total of 46 features, including the target variable.

· IoT-23 [40] contains captured real traffic by the avast AIC laboratory in partnership with the Czech technical university in Prague. IoT-23 contains twenty malware captures from different IoT devices and three captures for benign anomalies.

· NSL-KDD [45] is an improved version of KDD99 and has evolved by eliminating redundant then duplicate records. In the present study, we have used 20% of NSL-KDD taking into account all features except the target vector.

The experimental evaluation of our approach is performed on multi-core Intel® Core™ i7-1165G7 @ 2.80 GHz. 2.80 GHz and GPU Nvidia® PhysX® GeForce

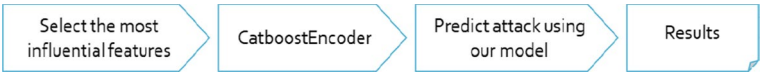

Fig. 3 Validation process

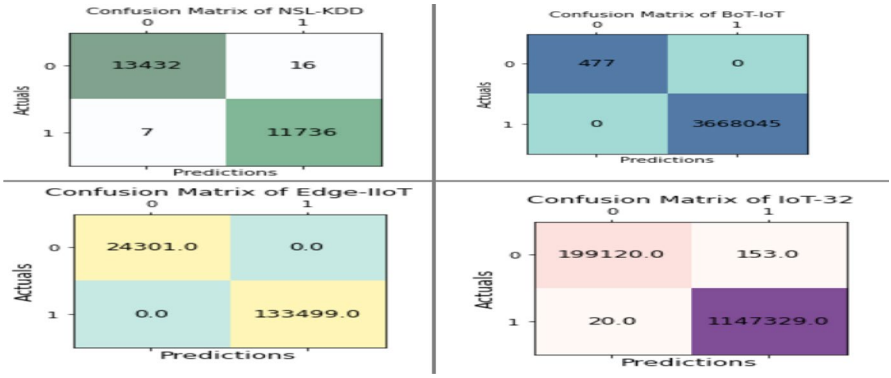

Fig. 4 Confusion matrix of prediction on BoT-IoT, IoT-23, NSL-KDD datasets

MX330 with 8 GB RAM and 64-bit operating system. The model is implemented using Jupyter Lab under python 3.9.7 and Catboost 1.0.3, including pandas, NumPy, sklearn libraries, and driver GPU.

- **In the learning phase,** we form the model following the steps described in Sect. 3.1; we partition our data into two portions of 50%-50%. In training, we used 200 iterations, trained the model using 50% of the dataset, and created two random permutations in Catboost. Furthermore, we used the gradient to calculate the values in leaves with three depth maximum. The founders of CatBoost are already testing this practice. As mentioned, K-fold when K = 2 is the best for most datasets since it does not suffer from conditional shift [44]. On the other hand, it will allow us to keep more data for prediction tests.
- **In the Validations phase,** following these steps in Fig. 3, we used all data (100%) to evaluate the model. Firstly, to select the most influential features, we used CatboostEncoder to deal with the categorical features; then, we used the model to predict the Attacks in all data.

4.2 Experimental results and discussion

- **Binary classifications:**

We use datasets according to our model, implementing the process steps defined in Fig. 2. Firstly, we pre-process the datasets and then define, extract, and encode features vector and target labels; we use all features in the first training. Subsequently, we must define train_size, test_size, and hyperparameter in Table 3 to train and test our model using the open-source plate-forme Catboost. In our experimentation, we

used 160 iterations and we created two random permutations of our training data. We used a gradient to calculate the values in leaves with three depth maximum. This operation's resulting complexity is O(2n). After testing, we obtain the following results (Fig. 4).

Using the BoT-IoT dataset, we obtain good results in Table 5 and Fig. 5, our model achieving the highest intrusion detection performance in accuracy, precision, and recall, around 100%. The confusion matrices shown in Fig. 4 describe that the model is successfully achieved with 0 FPR and 0 FNR. Figure 6 describes the learning and detection time. It needs seven iterations with a performance time of 4,25 s to fit the model on GPU and 0,865 s to detect attacks in all data. For those successful results, we used 42 features to train and test our model, but we used just 25 features in validation that contribute to the performance, as presented in Fig. 7 and Table 6.

Table 5 Performance measures result on Edge-IIoT, BoT-IoT, IoT-23, NSL-KDD

	Accuracy%	Precision%	Recall %	Learning time(s)	Detection time(s)
Edge-IIoT	100	100	100	1	0,146
BoT-IoT	100	100	100	4,25	0,865
NSL-KDD	99,81	99,72	99,88	3,58	0,108
IoT-23	99,98	99,98	99,99	12,3	0,763

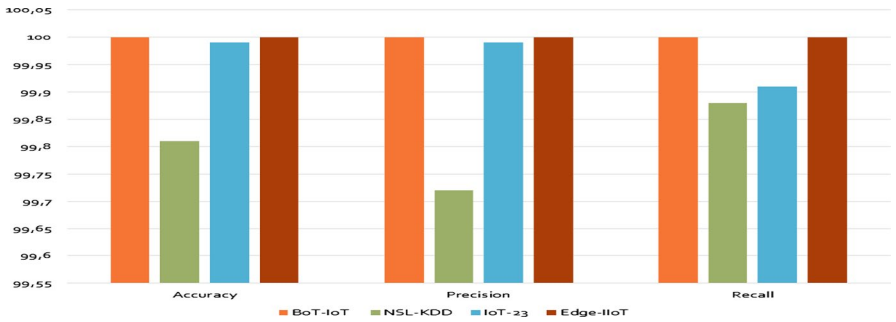

Fig. 5 Performance evaluation of our model

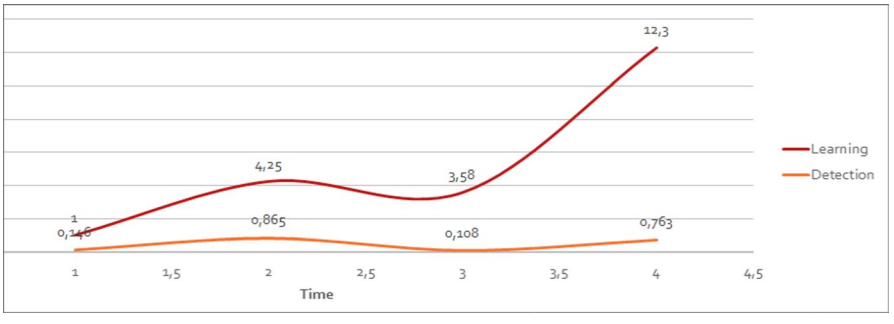

Fig. 6 Learning time and detection time of different datasets

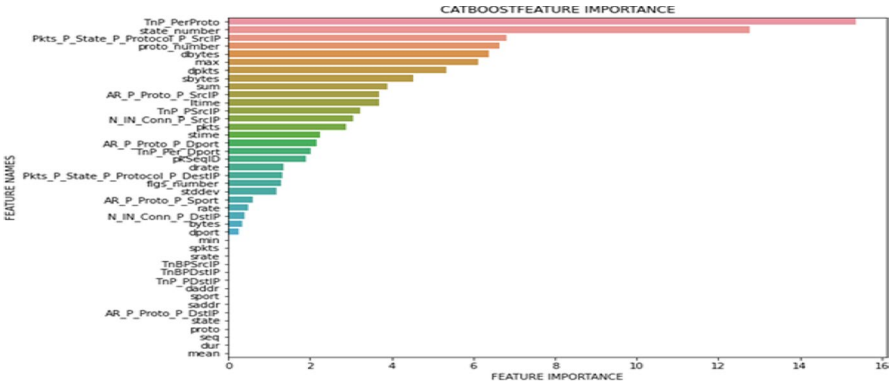

Fig. 7 Influential features of detection attack on BoT-IoT dataset

Table 6 BoT-IoT Features used in validation

Features	Description
Tnp_PerProto	Total Number of packets per protocol
Pkts_P_State_P_Protocol_P_SrcIP	Number of packets grouped by state of flows and protocols per source I
proto_number	Numerical representation of feature proto
dbytes	Destination-to-source byte count
max	Maximum duration of aggregated records
sbytes	Source-to-destination byte count
dpkts	Destination-to-source packet count
sum	Total duration of aggregated records
AR_P_Proto_P_SrcIP	Average rate per protocol per Source IP. (calculated by pkts/dur)
Itime	Record last time
TnP_PSrcIP	Total Number of packets per source IP
N_In_Conn_P_SrcIP	Number of inbound connections per source IP
pkts	Total count of packets in transaction
stime	Record start time
AR_P_Proto_P_Dport	Average rate per protocol per dport
Tnp_Per_Dport	Total Number of packets per dport
Drate	Destination-to-source packets per second
Pkts_P_State_P_Protocol_P_DestIP	Number of packets grouped by state of flows and protocols per destination IP
flgs_number	Numerical representation of feature flags
stddev	Standard deviation of aggregated records
AR_P_Proto_P_Sport	Average rate per protocol per dport
rate	Total packets per second in transaction
N_IN_Conn_P_DstIP	Number of inbound connections per destination IP
bytes	Total number of bytes in transaction
dport	Destination port number

We use IoT-23 to Compare and confirm the performance of the model. The obtained results confirm the performance of the model. In Table 5, Fig. 5, all accuracy, precision, and recall results are around 99.9%. Conversely, the error is minimal and converges to zero with 0.00002 FNR and 0.00018 FPR, as shown by the confusion matrix in Fig. 4. Moreover, Fig. 6 shows that we need just 12 s to fit the model in GPU and 0,763 s to detect attacks in all dataset. In addition, we used 18 features for those successful results, but only 12 influenced detection features. We use 20% NSL-KDD to confirm and compare the performance of our model. As shown in Table 5, Fig. 4, and Fig. 5, our model still performs well. The best iteration is in a total time of 3,85 s to fit and a detection time of just 0,108 s.

Again, as discussed above, the results obtained confirm the model's performance in accuracy, precision, and recall with 99.8% and 0.00068 FPR, and 0.00082 FNR. We tested the model with Edge-IIoT and obtained higher results in Table 5. The obtained results confirm the performance of the model. All accuracy, precision, and recall results are 100%. On the other side, the error is zero with 0 FNR and 0 FPR, as shown by the confusion matrix in Fig. 4. Moreover, Fig. 6 shows that we need just 1 s to fit the model in GPU and 0,146 s to detect attacks in all datasets.

- **Multiclass classification in Edge-IIoT:**

The result of categorical classification in Edge-IIoT, as shown in Table 7 and Fig. 8, proved that the model produced a detection rate comparable to that of the binary classification model. The model has a comparatively high level of precision and accuracy throughout training, 100%, and validation 99,27%.

FPR and FNR rates in the model are meager. Additionally, as shown in Fig. 9, it has a higher detection rate for the normal classes and malicious classes like DDoS ICMP, DDoS UDP, MITM, Password, SQL injection, Uploading, and

Table 7 Performance measures of categorical classification result on Edge-IIoT

	Accuracy%	Precision%	Recall %
Backdoor	100	96	98
DDoS_HTTP	99	97	98
DDoS_ICMP	100	100	100
DDoS_UDP	100	100	100
Fingerprinting	100	86	92
MITM	100	100	100
Normal	100	100	100
Password	100	100	100
Port_Scanning	99	100	99
Ransomware	96	100	98
SQL_injection	100	100	100
Uploading	100	100	100
Vulnerability_scanner	100	100	100
XSS	97	99	98

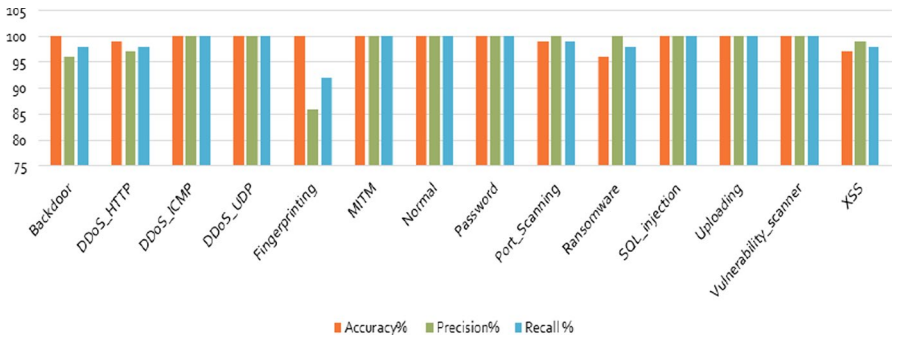

Fig. 8 Performance evaluation of the model on Edge-IIoT

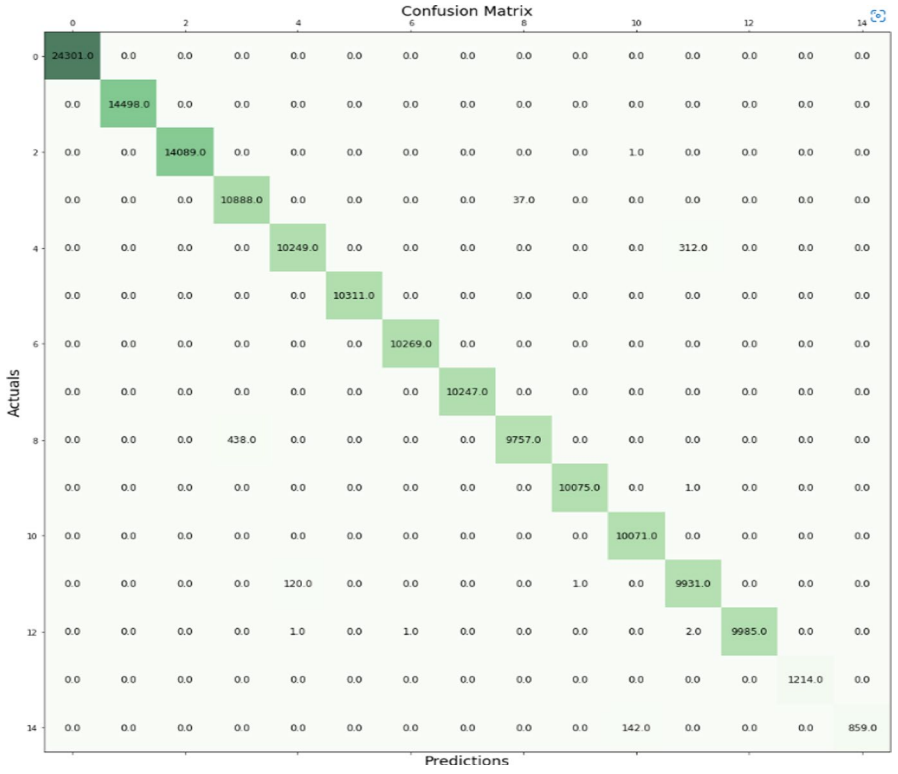

Fig. 9 Confusion matrix of multiclass prediction on Edge-IIoT

Vulnerability Scanner with 100% of precision and recall compared to other malicious classes like DDoS HTTP, Port Scanning, Ransomware, Backdoor, Fingerprinting, and XSS that has recalled around of 98%. Moreover, the model performs well in record time in terms of detection time with 0,44 s in all data. Our research

leads us to conclude that the model is still performant and identifies abnormalities in multiclass classification.

The binary and multiclass classifications were performed using a Catboost, especially gradient boosting for decision trees trained and validated on GPU. The model took less time to train between 1 and 12 s. It took a record to validate between 0,1 and 0,8 s. The training model then validates it using the influence features, further reducing the calculation time and increasing IDS' accuracy and precision. The model benefitting from TS using gradient boosting and Catboost-Encoder deal with the huge volume and solve class imbalance by optimizing the detection of minority classes using target statistics and gradient boosting. Furthermore, the use of GPU benefits and influences this model.

We tested the model on different datasets to make a comparison. The model proved to be fast and had a good detection rate. So, integrating GPU at the fog computing level can potentially minimize the intrusion detection time and assure responsiveness. According to the performance comparison presented in Table 8

Table 8 Comparison of some intrusion detection methods on BoT-IoT, IoT-23, NSL-KDD datasets

	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	Dataset
Shafiq et al. [37] (2020)	BayesNet	99.77	100	99	BoT-IoT
	C4.5	99.99	100	100	
	NaiveBayes	99.79	99	98	
	RandomForest	99.99	100	100	
	RandomTree	99.99	100	100	
Ullah et al.[12]. (2021)	CNN 3D	99.90	99.75	99.85	
Koroniotis, et al.[32]. (2016)	SVM	99.99	99.99	100	
	RNN	97.90	99.99	97.90	
	LSTM	98.05	99.99	98.05	
Our Model	Catboost	100	100	100	
Revathi and Malathi [38] (2013)	J48graft	99.57	96.8	96.9	NSL-KDD
	C4.5	99.55	97.1	97.1	
J. Gu et al. [4] (2020)	NB-SVM	99.35	–	99.24	
Tama et al.[22] (2017)	GBM	99.85	–	–	
Li et al.[41] (2018)	BC + k-NN	94.92	98.72	92.28	
Primartha et al. [27] (2017)	Random Forest	91.8	–	–	
Koroniotis et al. [33] (2020)	SwiftIDS	99.67	99.7	99.59	
Guezzaz et al. [47] (2021)	DT + Enhanced Data Quality	99.42	–	98.20	
Prabavathy et al. [21] (2018)	Deep model	99.20	–	99.27	
	Shallow model	95.22	–	97.50	
Our model	Catboost	99.92	99.88	99.92	
Ullah et al [12]. (2021)	CNN3D	99.98	99.90	99.98	IoT-23
Stoian [39] (2020)	ANN	66	71	66	
	RF	99.5	99.5	99.5	
	AdaBoost	87	86	87	
Our model	Catboost	99.99	99.99	99.99	

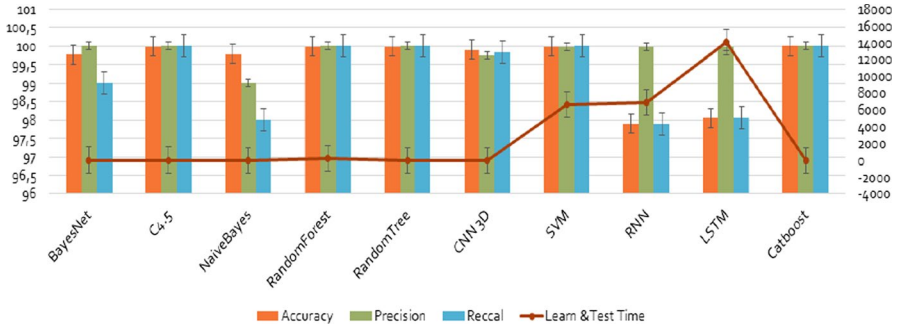

Fig. 10 Comparison of performance and processing time of ML and DL intrusion detection

and Fig. 10, our proposed model achieves the highest performance and outperforms all other IDSs in this study in terms of robustness and time performance.

5 Conclusion and future work

Intrusion detection is ideal for reinforcing IoT security against attacks, especially when integrating IDS in fog computing. This paper presents an optimized intrusion detection model for IoT security based on an anomaly detection method to enhance IDS accuracy with time processing performance. The results of the experiments realized on multiple datasets, and the performance comparisons that have been made have proven that our model is the highest and most robust performance with the lowest cost in time. The model benefits from TS using gradient boosting and CatboostEncoder that deal with the huge volume and solve class imbalance by optimizing the detection of minority classes using target statistics and gradient boosting. The use of GPU benefits and influences the model. According to this study, the suggested model would contribute to developing an efficient IoT network intrusion detection system with a high detection rate. In addition, this work confirms that Catboost is a powerful ML. For future work, we plan to use Blockchain enhancement with machine learning methods to reinforce security in IoT environments.

Funding Our work has not been funded and has been worked without financial support. We did this research work as professors of computer science at the university.

Data availability Assessments and experimental results, obtained using Anaconda 3 IDE, are available and will be shared with authors at https://sites-Google.com/umi.ac.ma/azrour.

Declarations

Conflict of interest We declare that we have no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by authors.

References

1. Alabaa FA, Othmana M, Hashema IBT, Alotaibib F (2017) Internet of Things security: a survey. J Netw Comput Appl 88:10–28
2. Noor M, Wan Hassan H (2018) Current research on Internet of Things (IoT) security: a survey. Comput Netw 148:283–294
3. Sethi P, Sarangi SR (2017) Internet of Things: architectures, protocols, and applications. J Electrical Comput Eng 2017:9324035
4. Nasir M, Javed AR, Tariq MA et al (2022) Feature engineering and deep learning-based intrusion detection framework for securing edge IoT. J Supercomput 78(6):8852–8866
5. Chanal PM, Kakkasageri MS (2020) Security and privacy in IoT: a survey. Springer Sci 115(2):1667–1693
6. Ferraga MA, Maglaras L, Moschoyiannis S, Janicke H (2020) Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. J Inf Secur Appl 50:102419
7. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J (2019) Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2(1):1–22
8. Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surveys Tutor 18(2):1153–1176
9. Fadlullah ZM, Tang F, Mao B, Kato N, Akashi O, Inoue T, Mizutani K (2017) State-of-the-art deep learning: evolving machine intelligence toward tomorrow‘s intelligent network traffic control systems. IEEE Commun Surveys Tutor 19(4):2432–2455
10. Da Costa KAP, Papa JP, Lisboa CO, Munoz R, de Albuquerque VHC (2019) Internet of Things: a survey on machine learning-based intrusion detection approaches. Comput Netw 151:147–157
11. Agrawal S, Agrawal J (2015) Survey on anomaly detection using data mining techniques. Procedia Comput Sci 60:708–713
12. Ullah I, Mahmoud QH (2021) Design and development of a deep learning-based model for anomaly detection in IoT networks. IEEE Access 9:103906–103926
13. Hodo E, Bellekens X, Hamilton A, Dubouilh PL, Iorkyase E, Tachtatzis C, Atkinson R (2016) Threat analysis of IoT networks using artificial neural network intrusion detection system. International Symposium on Networks, Computers and Communications (ISNCC). IEEE, Yasmine Hammamet, pp 1–6
14. Hosseinpour F, Vahdani Amoli P, Plosila J, Hmlinen T, Tenhunen H (2016) An intrusion detection system for fog computing and IoT based logistic systems using a smart data approach. Int J Digit Content Technol Appl 10(5):34–46
15. Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P (2018) Network intrusion detection for IoT security based on learning techniques. IEEE Commun Surveys Tutor 21(3):2671–2701
16. Sheng Z, Yang S, Yu Y, Vasilakos A, Mccann J, Leung K (2013) A survey on the IETF protocol suite for the internet of things: standards, challenges, and opportunities. IEEE Wirel Commun 20(6):91–98
17. Zeng D, Guo S, Cheng Z (2011) The web of things: a survey. J Commun 6(6):424–438
18. Chaabouni N, Mosbah M, Zemmari A, Sauvignac C (2020) A OneM2M intrusion detection and prevention system based on edge machine learning. IEEE/IFIP Network Operations and Management Symposium. IEEE, Budapest, pp 1–7
19. Verma A, Ranga V (2019) Machine learning based intrusion detection systems for IoT applications. Springer Sci Bus Media 111(4):2287–2310
20. Diro AA, Chilamkurti N (2017) Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener Comput Syst 82:761–768
21. Prabavathy S, Sundarakantham K, Shalinie SM (2018) Design of cognitive fog computing for intrusion detection in Internet of Things. J Commun Netw 20(3):291–298
22. Tama BA, Rhee KH (2017) An in-depth experimental study of anomaly detection using gradient boosted machine. Neural Comput Appl 31(4):955–965
23. Ferrag MA, Friha O, Hamouda D, Maglaras L, Janicke H (2022) Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access 10:40281–40306. https://doi.org/10.1109/ACCESS.2022.3165809
24. Misra S, Krishna PV, Agarwal H, Saxena A, Obaidat MS (2011) A learning automata based solution for preventing distributed Denial of Service in Internet of Things. IEEE International Conferences on Internet of Things, and Cyber Physical and Social Computing. IEEE, Dalian, pp 114–122

25. Kasinathan P, Pastrone C, Spirito MA, Vinkovits M (2013) Denial-of-Service detection in 6LoWPAN based Internet of Things. IEEE 9th International Conference on Wireless and Mobile Computing Networking and Communications. IEEE, Lyon, pp 600–607
26. Sonar K, Upadhyay H (2016) An Approach to Secure Internet of Things against DDoS In: Proceedings of International Conference on ICT for Sustainable Development, Springer science business media, Singapore, pp. 367–376 DOI: https://doi.org/10.1007/978-981-10-0135-2_36
27. Primartha R, Tama BA (2017) Anomaly detection using random forest: a performance revisited. International Conference on Data and Software Engineering. IEEE, Palembang, pp 1–6
28. Raza S, Wallgren L, Voigt T (2013) SVELTE: Real-time intrusion detection in the Internet of Things. Ad Hoc Netw 11(8):2661–2674
29. Cervantes C, Poplade D, Nogueira M, Santos A (2015) Detection of sinkhole attacks for supporting secure routing on 6lowpan for Internet of Things. IFIP/IEEE International Symposium on Integrated Network Management. IEEE, Curitiba, pp 606–611
30. Bostani H, Sheikhan M (2020) Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on mapreduce approach. Comput Commun 98:52–71
31. Al-kasassbeh M, Abbadi MA, Al-Bustanji AM, ightGBM Algorithm for malware detection‖ In: Intelligent Computing, pp. 391–403 https://doi.org/10.1007/978-3-030-52243-8_28
32. Guezzaz A, Asimi A, Sadqi Y, Asimi Y, Tbatou Z (2016) A new hybrid network sniffer model based on PCAP language and sockets (PcapSockS). Int J Adv Comput Sci Appl (IJACSA), 7(2) DOI https://doi.org/10.14569/IJACSA.2016.070228
33. Jin D, Lu Y, Qin J, Cheng Z, Mao Z (2020) SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism. Comput Secur 97:101984
34. Adebowale A, Idowu S, Amarachi AA (2013) Comparative study of selected data mining algorithms used for intrusion detection. Int J Soft Comput Eng 3(3):237–241
35. Thaseen S, Kumar CA (2013) An analysis of supervised tree based classifiers for intrusion detection system. International Conference on Pattern Recognition, Informatics and Mobile Engineering. IEEE, Salem, pp 294–299
36. Hancock JT, Khoshgoftaar TM (2020) CatBoost for big data: an interdisciplinary review. J Big Data 7:94. https://doi.org/10.1186/s40537-020-00369-8
37. Shafiq M, Tian Z, Sun Y, Du X, Guizani M (2020) Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. Futur Gener Comput Syst 107:433–442
38. Revathi S, Malathi A (2013) A Detailed Analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. Int J Eng Res Technol 2(12):1848–1853
39. Stoian, N.A. (2020) Machine learning for anomaly detection in IoT networks: Malware analysis on the IoT-23 data set—University of Twente, Student Theses (utwente.nl)
40. Garcia S, Parmisano A, Erquiaga MJ (2020) IoT-23: a labeled dataset with malicious and benign IoT network traffic (version 1.0.0). Zenodo. 10.5281/zenodo.4743746
41. Li L, Yu Y, Bai S, Hou Y, Chen X (2018) An effective two-step intrusion detection approach based on binary classification and k-NN. IEEE Access 6:12060–12073
42. Anna Veronika Dorogush, Vasily Ershove, and Andrey Gulin (2018) CatBoost: gradient boosting with categorical features support, arXiv:1706.09516v5.
43. Abdullahi A. Ibrahim, Raheem L. Ridwan, Muhammed M. Muhammed, Rabiat O. Abdulaziz and Ganiyu A. Saheed (2020) Comparison of the CatBoost classifier with other machine learning methods. Int J Adv Comput Sci Appl (IJACSA), 11(11) DOI: https://doi.org/10.14569/IJACSA.2020.0111190.
44. Prokhorenkova L, Gusev G, Vorobev A, Dorogush A, Gulin A (2018) CatBoost: Unbiased Boosting with Categorical Features. Proceedings of the 32nd International Conference on Neural Information Processing Systems 31:6639–6649
45. Tavallaee M, Bagheri E, Lu W, Ghorbani A (2009) A detailed analysis of the KDD CUP 99 Data Set. IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA). IEEE, Ottawa, pp 1–6
46. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B (2019) Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Futur Gener Comput Syst 100:779–796
47. Guezzaz A, Benkirane S, Azrour M, Khurram S (2021) A reliable network intrusion detection approach using decision tree with enhanced data quality. Secur Commun Netw 2021(8):1230593

48. Azrour M, Mabrouki J, Guezzaz A, Kanwal A (2021) Internet of Things security: challenges and key issues. Secur Commun Netw 11:5533843
49. Guezzaz A, Asimi Y, Azrour M, Asimi A (2021) Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection. Big Data Min Anal 4(1):18–24
50. Azrour M, Mabrouki J, Chaganti R (2021) New efficient and secured authentication protocol for remote healthcare systems in Cloud-IoT. Secur Commun Netw 4:1–12
51. Guezzaz A, Asimi Z, Batou Y, Asimi Y, Sadqi Y (2019) A global intrusion detection system using pcapsocks sniffer and multilayer perceptron classifier. Inter J Netw Secur 21(3):438–450
52. Guezzaz A, Asimi A, Asimi Y, Tbatou Z, Sadqi Y (2017) A lightweight neural classifier for intrusion detection. General Lett Math 2(2):57–66
53. Idhammad M, Afdel K, Belouch M (2018) Semi-supervised machine learning approach for DDoS detection. Appl Intell 48:3193–3208
54. Kaja N, Shaout A, Ma D (2019) An intelligent intrusion detection system. Appl Intell 49:3235–3247
55. Çavuşoğlu Ü (2019) A new hybrid approach for intrusion detection using machine learning methods. Appl Intell 49:2735–2761
56. Kumar G (2020) An improved ensemble approach for effective intrusion detection. J Supercomput 76:275–291

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.